TITLE EXECUTIVE - ALP 1

SHEET OF SHEETS

# USER SPECIFICATION

## EXECUTIVE - ALP 1

|  |  |
|---|---|
| AMD'T RECORD | A |

This drawing and any information or descriptive matter set out hereon are the confidential and copyright property of INFORMATION COMPUTER SYSTEMS LTD and must not be disclosed loaned copied or used for manufacturing, tendering or for any other purpose without their written permission

# INDEX

# 1.Ø. IDENTIFICATION

## 1.1. Title

Executive - ALP 1

Catalogue Number 774ØØØ

## 1.2. Function

The Executive program provides the working environment for all the programs in the system and administers level changing and program communication.

## 1.3. Operating Environment

The Executive operates on ALP 1 only, and requires a Multi-Level Interrupt Unit (MIU) to be present.

## 1.4. Availability

The program is available on paper tape in source code form, catalogue number 774ØØØ SM ØØØ.

## 2.Ø. • GENERAL DESCRIPTION

The Executive program controls the system on-line by responding to interrupts which can be generated by user program or by hardware in the system. When no program is active Executive enters a 'WAIT' state. Executive always enters the highest priority level active at any time, lower levels being entered when the higher level has completed its task. The machine registers belonging to each program are always preserved by Executive when a change of level occurs and,therefore,the changes in level are not normally seen by the user program.

Executive handles communication between the programs in the system by means of a queueing technique. Each message has a priority associated with it and messages are queued in priority order, messages of the same priority being queued in chronological order.

User programs communicate with Executive by means of Executive Calls, (details of which are given below) thereby requesting Executive to perform certain functions for them.

Executive also takes appropriate action in the case of catastrophic machine failure, or individual program failures. In the case of machine failures the machine is halted after an appropriate message has been output. Individual programs which fail are inhibited from further action.

In the remainder of this document it is assumed that ALP 1 /ALP 2 compatability is required. If this is not the case then the segment structure need not be adhered to, as on ALP 1 absolute addresses are used throughout.

## 2.1. Terminology.

### 2.1.1. Domain

Each program has available to it a maximum of 16 domains, each consisting of up to 4K words of store.
Each domain has associated with it a status, set at Integration time, which can only be altered by privileged programs.
The status available are 'not available', 'code only','read only', and 'read/write'.

### 2.1.2. Segment

A segment is a continuous (in store) part of a program, occupying one or more domains all of which must have the same status.
Thus a program can be considered to consist of a collection of segments.
Although only the segments currently occupying the 16 domains are available to a user at any one time the program may call for another segment (normally held on backing store) to occupy any domain(s) except domain 1 (see below). The program cannot then access the previous contents of that domain.

For ALP 1 only, program segments are called down to pre-specified addresses and there is therefore only that loss of access which results from the overlay structure.

### 2.1.3. Module

Segments are comprised of modules which are merely convenient sub-divisions of the segment. A segment may consist of any number of modules.

### 2.1.4. Program

A program consists of a collection of segments, occupies a single priority level, and has a Master Segment associated with it.

### 2.1.5. Master Segment

Every program has associated with it a Master segment normally produced at Integration time. The Master segment defines the program and its requirements to the Executive. The Master segment is not normally referenced by the user program, but always occupies domain one of the program. For full details of the Master Segment Structure, see Appendix 1.

### 2.1.6. Job Segment

Every program has a six word area of core associated with it, normally produced at Integration time, which always occupies domain zero. These locations are overwritten with the program's next queue item when the program makes an appropriate Executive Call. The format of queue items is at the user's discretion except that:-

1) The third word of the queue item is always cleared before the item is sent so that it can be used for status reporting.

2) The first word of the queue item cannot have all ones in it, if an attempt is made to send such an item it will be returned with illegal command set in the status word.

3) Bit three of the first word of the queue item must always be zero for non-privileged mode programs.
It should also be noted that if any of the three most significant bits of the first word of the item are non-zero then the queue item is assumed to have data associated with it, the address of which is specified in the second word of the queue item. This address will be checked to see if it lies within the program overlay area (see section 3.2.4.) and, if so, when the queue item becomes the receiving level's current item the sending level will be brought into the overlay area, and remain until the item is returned.

### 2.1.7. Priority Level

Each program is associated with a unique priority level which determines which of the programs active at any one time is to be obeyed. Priorities are numbered from zero (highest priority) to 255 (lowest priority), but level zero is normally reserved for the Time program.

### 2.1.8. Communication Channel

Each program has associated with it one or more communication channels which are specified in the Master Segment and are normally set up at Integration Time.

These specify the programs (or, in the case of Device Routines, IOP levels as well) with which communication is allowed, and the priority of the communication.

### 2.1.9. Message Priority

Every message sent through the system has a priority associated with it. Messages sent to a program are queued in priority order, messages of equal priority being queued in chronological order. Priorities zero to fifteen are allowed, priority fifteen being reserved for sending a message to another program with the same priority as the sending program's current queue item.

# 3.Ø UTILIZATION

## 3.1. Data Lists

### 3.1.1. Master Segment

Every program level has associated with it a master segment. This is normally produced at Integration time, and contains data on the program's communication channels. It is also used for storage of the machine registers when the level is not active. For full details of the method of production of the master segment see the User Specification of Integrator.

The format of the Master Segment is given in Appendix 1.

### 3.1.2. Program Status Definition (PSD) List.

This is a list of one item per program leve and is used by Executive to determine the current status of the level. A detailed account of its structure is given in Appendix 2.

### 3.1.3. Queue Definition List

This is a list of one item for each program level plus one item for each IOP level. In the case of the BIOC each device is considered to be on a separate level and consquently has a separate entry in the list. The format of the entries is given in Appendix 3.

### 3.1.4. Queue Area

This is the area in which Executive queues the communication requests to all program levels and IOP levels. In order to reduce the size of this area each queue is a chained list, and queue zero contains the chain of unused items-hence no item may be sent to program level zero. The format of items in the queue area is given in Appendix 4.

### 3.1.5. System Record Area

This is only included if the System Record Option is included. The contents of the area are items loaded by Executive whenever a level is entered, becomes active when inhibited, or makes an Executive Call. The area has associated with it parameters which define the start and end of the area, and the current oldest item (the area is used in a cyclic fashion). The complete structure is defined in Appendix 5.

It should also be noted that there exists a utility program which generates an interpretive output of this area in chronological order. For details of this see System Record Print Utility Specification.

### 3.1.6. Backing Store Directory

This is a list of four words per entry,one entry being required for each segment that is backing store resident and one for each backing store resident program.

In addition one extra reference is required for each program or segment which has a clean copy on a backing-store.

## 3.2. Program Structure

A program consists of various segments , as illustrated below:

One and only one Master Segment
One and only one Job Segment
One or more Code Segments.
Read only Data Segments
Read/Write Data Segments

Any number of Data Segments of either type is allowed, except that the total number of segments a program can have is 256 of all types. The functions of the Master Segment and Job Segment have been defined above (sections 2.1.5. and 2.1.6.).

### 3.2.1. Code Segment

These should contain code and possibly some constants only. Code cannot appear in any other type of segment, and Code Segments can only be obeyed or read from.

### 3.2.2. Data Segments

Read only Data Segments should only contain data which the program does not change. Either type of Data Segment may be used for sending messages, as the segment will be made read/write to the receiving program. The message parameters (or queue item) should always be at the start of the segment and the segment must only occupy one Domain. All the data required by the receiving program must be contained within the Segment. When the message is sent the sending program cannot access the segment until it has been returned.

### 3.2.3. Segment Overlay

Facilities are provided via Executive and the Space Administration program for programs to reduce their effective size by means of a pre-planned Segment Overlay Scheme. The program requests segments to be brought into core store when required. For further details see the Space Administration Specification.

### 3.2.4. Program Overlay

Executive and Space Administration also allow a pre-planned program overlay structure to be used thereby reducing the total core size required for a complete system. This overlay is invisible to the programs operating under it except that data that passes between two overlaying programs must not be inside the Overlay Area. For further details of the Program Overlay facilities see Space Administration Specification.

### 3.2.5. Combined Program and Segment Overlay

It is also possible to have programs in the Program Overlay Area which also have their own Segment Overlay Structure as well, the Segment Overlay Area being either inside or outside the Program Overlay Area, although if inside the Program Overlay Area it must be continuous with the rest of the program. Full details are contained in the Space Administration Specification.

## 3.3. Program Communication

Messages are sent from program to program via Executive's queueing routines and the message must always be in the first six words of the segment specified, the data being anywhere else in the same segment. The segment must only occupy one domain (i.e. occupy less than 4K words of store). The ALP 1 Executive takes the six word message and copies it into the receiving level's domain zero, however on ALP 2 and ALP 3 the whole domain will be given to the receiving level as its domain zero and the sending level will no longer be able to access it. Executive Calls (see below) are available to specify whether or not the message is to be returned. If it is returned then whatever is in the receiving level's domain zero is returned to the original sending level.

### 3.3.1. Allocation

In order to prevent undesirable interleaving of messages,facilities exist in Executive to allocate a queue to one particular program and then de-allocate it when the set of messages is complete. Any other program attempting to send messages to the program whose queue is allocated is de-activated until the queue has been de-allocated.

However programs which have urgent message can have a channel specified as not held by allocation in which case its message will always be inserted in the queue regardless of its allocation.

### 3.3.2. Message Limit

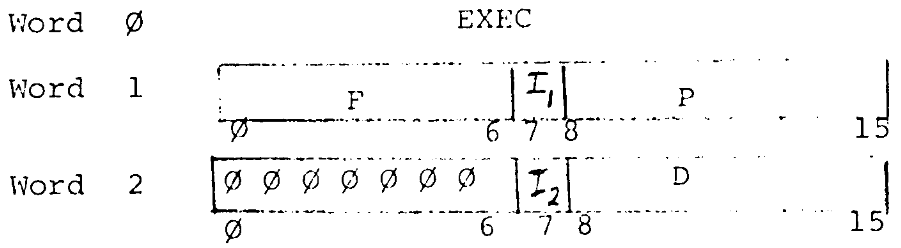
There is a limit (specified in the program Queue Definition and set up at Integration time) to the number of messages a program can have outstanding at any time. This is the number of messages which have been accepted into the queue system by Executive and have not yet been returned by the receiving level(s). If a program attempts to send a message which would cause it to exceed this limit then it is de-activated until one of its outstanding messages is returned.

### 3.3.3. Program Time-Out

It is possible to specify (at Integration time for inclusion in the program's Queue Definition) a time limit for the program in dealing with each message. If this time limit is exceeded then the program is inhibited from further action. A time limit of zero means that the program has unlimited time for dealing with each message. It should be noted that the time specified will be the maximum time allowed, and the actual time for any particular message can be up to one second less, due to the relationship between when the level receives the message and when the System Timer interrupt occurs. For further details see Time Program Specification.

## 3.4. Executive Calls

Programs communicate with the Executive by means of Executive Calls. These consists of three succesive locations in store, with the following format.



Word 1, byte Ø to 7 are referred to as the function byte.

F is the Executive Call function.
$I_1$ is the indirect marker for the parameter
P is the displacement of the Executive Call parameter
$I_2$ is the indirect marker for the data
D is the displacement of the Executive Call Data.

Both displacements are relative to the register P and are eight bit signed values. The indirect bits, if set indicate that the address of the required data is to be found in the location specified, not the data itself.

Channel and segment numbers are always obtained by using the least significant seven bits of the word specified.

Re-entry to the program is usually to the next location in store, with class 2 interrupts always in the same stat as before the Call.

The full range of Executive Calls are given below in basic format. The format for the Calls in normal (macro) form is given in Appendix 6.

### 3.4.1. Exit from Program

Function byte:- [ØØ] direct [Ø1] indirect

Parameter. This is the number of seconds that can elapse before the program is re-activated. If the value is zero then the Executive System will not re-activate the program until required to do so by another program.

Data This call does not require any data.

After obeying this Executive Call the program is not re-entered immediately, but may be re-entered whenever required. In any case the program will be activated after the delay specified, provided this is not zero, and the Time program is included in the system. It should be noted that if a delay of $x$ seconds is specified then the level will be activated between $(x-1)$ and $x$ seconds after the Executive Call, depending on the chronological relationship between the next system timer interrupt and the Executive Call.

### 3.4.2. Abandon Program

Function Byte: [Ø8] direct [Ø9] indirect

Parameter As for Exit from Program (3.4.1)

Data This Call does not require any data.

This Executive Call is normally only used on Systems employing Space Administration.

In other systems its action is identical with Exit from Program (3.4.1.). If Space Administration is used, and the program using this Call is backing -store resident, then the old copy of this program will not be dumped, and when the program is re-entered a clean copy of the program will be brought down from the backing store. Otherwise the action is identical with Exit from Program (3.4.1.). Using this Call wherever possible increases the system's efficiency by reducing the number of backing store transfers necessary. It should also be noted that programs in the Program Overlay Area will obtain clean copies of each segment when they are called.

## 3.4.3. Request Date.

Function byte: [Ø4] direct [Ø5] indirect

Parameter — This Call does not require any Parameter

Data — The address of a three word buffer into which the date is to be copied.

The program is re-entered immediately, provided it is still the highest priority level or had class 2 interrupts inhibited, with the current values of the day, month and year respectively in the buffer area specified. The values are not in character form, and the full value is given for the year (e.g. 1972 not 72).

### 3.4.4. Request Time

Function byte: [Ø6] direct [Ø7] indirect

| | |
|---|---|
| Parameter | This call does not require any Parameter. |
| Data | The address of a three word buffer into which the time is to be copied. |

The program is re-entered immediately, provided it is still the highest priority level or had class 2 interrupts inhibited, with the current values of the hour, minute and second respectively in the buffer area specified. The values are not in character form and the hour is given on a 24-hour clock basis.

### 3.4.5. Activate Another Program

Function byte: [1Ø] direct [11] indirect

| | |
|---|---|
| Parameter | The channel number which refers to the program to be activated. |
| Data | This Call does not require any data. |

The program is re-entered immediately, provided it is still the highest priority level or had class 2 interrupts inhibited.

The program requested is activated and entered as soon as it becomes the highest priority level, and interrupts are allowed. This can be before the activating program is re-entered. Whereas when sending a message (see sections 3.4.8., 3.4.9. 3.4.10.and 3.4.11.) to another program, channel zero can be used, this is invalid for the activate Call. Specifying level zero in the channel data is accepted however, this normally being the Time program.

### 3.4.6. Inhibit Entry to a Program

Function byte: [ 18 ] direct [ 19 ] indirect

Parameter The channel number which refers to the program to be inhibited.

Data This Call does not require any data.

The program is re-entered immediately, provided it is still the highest priority level, or had class 2 interrupts inhibited. The program inhibit bit is set in the P.S.D. of the program specified preventing it from being entered until it has been cleared by TESTAID or an Allow Entry Executive Call (section 3.4.7.). Channel and level zero are treated as in section 3.4.5.

### 3.4.7. Allow Entry to a Program

Function byte: [ 2Ø ]direct [ 21 ] indirect

Parameter The channel number which refers to the program to be allowed.

Data This Call does not require any data.

The program is re-entered immediately, provided it is still the highest priority level, or had class 2 interrupts inhibited. The program inhibit bit is cleared in the P.S.D. of the program specified. This allows the program to be entered when next activated, provided no other condition exists to inhibit it. If an attempt was made to activate the program while it was inhibited then it will be activated when the 'Allow Entry' Call is made. Channel and level zero are treated as in section 3.4.5.

### 3.4.8. Send Message

Function byte: [ 4Ø ]direct[ 41 ]indirect

Parameter The channel number which refers to the Program to which the message is to be sent.

Data The address of the segment whose contents are to be sent.

The program is normally re-entered immediately, provided it is still the highest priority level, or had class 2 interrupts inhibited. However there are two circumstances when the program will be forcibly de-activated by Executive. If the channel on which the message is to be sent is already allocated to another program then the requesting program (i.e. the program making the Call) will be de-activated, and not re-activated until the channel is de-allocated. Any other program wishing to send a message on the same channel will also be re-activated at that time, and so the requesting program will have to wait until it is the highest priority program wishing to communicate with that receiving program. This assumes the requesting program's channel data does not have allocation over-ride set.

The other circumstance when the requesting program is de-activated is when the Queue area is full. The program is re-activated as soon as there is space available in the Queue Area. This is an abnormal situation caused by system overload.

The first six words of the segment specified are given to the receiving program (via the Queue area) as its job segment, when that message becomes the requesting program's current queue item. The channel is de-allocated and the message is not returned to the requesting level. Word 3 of the message is always set to zero, ready for status reporting.

If the requesting program specifies its job segment as the segment to be sent then the program's current queue item is sent on the channel specified and the user's program re-entered without being given the next queue item.

If the requesting program specifies channel zero, then the segment specified is abandoned, and the clean copy used when next requested.

3.4.9. Send Message and Allocate

Function byte: [42] direct [43] indirect

Parameter The channel number which refers to the program to which the message is to be sent.

Data The address of the segment whose contents are to be sent.

This Call is the same as the Send Message Call (3.4.9.) except that the channel specified is allocated to the requesting program.

3.4.10. Send Message and Return

Function byte: [44] direct [45] indirect

Parameter The channel number which refers to the program to which the message is to be sent.

Data The address of the segment whose contents are to be sent.

This Call is the same as the Send Message Call (3.4.9) except that the message is returned to the requesting program's queue when the receiving program Calls for its next queue item.

3.4.11. Send Message, Return and Allocate

Function byte: [46] direct [47] indirect

Parameter The channel number which refers to the program to which the message is to be sent.

Data The address of the segment whose contents are to be sent.

This call is the same as the Send Message Call (3.4.9.) except that the message is returned to the requesting program's queue when the receiving program calls for it's next queue item and the channel specified is allocated to the requesting program.

### 3.4.12. Hold for Queue Item

Function byte: [48]direct [49] indirect

Parameter The channel number which refers to the program from which an item is required.

Data The address of the segment to which the item is to be loaded or the data location into which the originating channel number is written.

The program is de-activated until the required queue item is loaded into its queue. In the case of a returned message then the required message is that which has come from the channel specified and has its word 1,the same as word 1 of the specified segment. If the item is not a returned item then any item from the channel specified is taken to be the required item. If the required item is in the queue then the item will be copied into the segment specified and the program re-entered immediately if it is still the highest priority level, or had class 2 interrupts inhibited.

When the required item is loaded into the 'HELD' level's queue from the channel specified then the item is copied directly into the segment specified and the level is re-activated.

If channel zero is specified then the level is assumed to be waiting for any queue item and the Call data points to a single location within the program's data areas. When an item is available then the channel number from which it came is written into the specified location so that the program may perform a 'HOLD' Call for that specific channel. If the program does not have a channel corresponding to the sending program, zero is written into the data location.

### 3.4.13. Test for Queue Available

Function byte: [5Ø]direct[51]indirect

Parameter The channel number which refers to the queue to be tested.

Data The return address if the queue is not available.

The program is re-entered immediately, provided it is still the highest priority level or had class 2 interrupts inhibited. If the channel specified cannot be allocated to the requesting level then the return is to the address specified by the data. If the channel specified is available then it is allocated to the requesting level, and return is to the location immediately after the Call.

Specifying channel Ø for this Call is invalid.

### 3.4.14. De-Allocate Channel

Function byte: [58]direct [59]indirect

Parameter The channel number which refers to the queue to be de-allocated.

Data This Call does not require any data.

The program is re-entered immediately, provided it is still the highest priority level or had class 2 interrupts inhibited. The channel specified is de-allocated, and any subsequent messages on that channel will have to await re-allocation of the channel.
Specifying channel Ø for this Call is invalid.

### 3.4.15. Call Segment, Abandoning Current Segments and Enter.

Function byte: [6Ø]direct [61]indirect

Parameter The number of the segment required by the program.

Data The address at which the program is to be re-entered.

The program is de-activated until the segment requested is available (if the segment is already available then the program is re-entered immediately provided it is still the highest priority level or had class 2 interrupts inhibited). The program is always re-entered at the address specified in the data.

As segments Ø to 15 are considered to be permanently available only segment numbers above 16 are valid, and the high limit is contained in the Master Segment for the program.

The program's current segments which overlay with the requested segment, if any, are not preserved in this call, i.e. are not written to dump areas on backing store and the next time they are requested clean copies will be provided.

If the job segment is specified then the next queue item for the program is inserted in its job segment and the previous item removed. In this case return is always to the location immediately after the call.

### 3.4.16. Call Segment, Dumping Current Segments and Enter.

Function byte: $\boxed{62}$direct $\boxed{63}$indirect

Parameter The number of the segment required by the program.

Data The address at which the program is to be re-entered.

This Call is similar to Call Abandon and Enter (3.4.15) except that the overlaying segments, if any, are preserved by being dumped on the backing store.

3.4.17. Call Segment, Abandoning Current Segments and Return.

Function byte: [64]direct [65]indirect

Parameter The number of the segment required by the program.

Data This Call does not require any data.

This Call is similar to Call, Abandon and Enter (3.4.15.) except that the program is always re-entered at the location immediately after the Call.

3.4.18. Call Segment, Dumping Current Segments and Return.

Function byte: [66]direct [67]indirect

Parameter The number of the segment required by the program.

Data This Call does not require any data.

This Call is similar to Call,Abandon and Enter (3.4.15) except that the overlaying segments, if any, are preserved by being dumped on the backing store and the program is always re-entered at the location immediately after the Call.

## 3.4.19. Test for Segment Available

Function byte [68]direct [69] indirect

Parameter — The number of the segment to be tested for availability.

Data — The return address if the segment is not available.

The program is re-entered immediately, provided it is still the highest priority level or had class 2 interrupts inhibited. If the specified segment is unavailable then the return is to the address specified by the data. If the segment is available then return is to the location immediately after the Call. This Call cannot be used for segments 1 to 15 as these are considered to be always available. If segment Ø is specified then the program's queue is tested to see if there is an item available, excluding any returned items.

## 3.4.2Ø. Abandon Segment.

This is achieved using a Send Message Call which specifies channel zero. See section 3.4.8.

## 3.5. Executive Entries

Executive is always entered as a result of an interrupt being generated. The action taken by Executive in each case is given below.

### 3.5.1. Power Failure Interrupt

The action taken by Executive depends on whether or not the 'Soft' Power Failure option is included. If it is not included then the Executive will halt inhibiting the system from further action. If the option is included then Executive will attempt to stop all IOP's and the BIOC normally (i.e. with a single TRGB instruction) the program then obeys a delay loop of instructions a user specified number of times. Any IOP (or the BIOC) still running is then crash-stopped and the machine halted. On power-up the system can be re-started by pressing 'run'. Any I-O transfer in progress when the power failure occurred will be returned with either 'stopped by TGRB' or 'crash-stop' set in its status.

### 3.5.2. Memory Parity Interrupt

The Executive immediately crash-stops the BIOC and outputs the message "MEMORY PARITY" to the teletype. The system is halted and can only be restarted by re-loading.

3.5.3. Invalid Function Interrupt.

The action taken by Executive depends on whether or not the 'All failures catastrophic' option is included. If it is included then a message of the form 'INVALID FUNCTION LXXX' (where XXX is the level causing the failure) will be output by Executive as in section 3.5.2. This option is intended for use with a fully proven system when any failure is catastrophic. If the option is not included then the offending level is inhibited from further action by the Invalid Function bit being set inits P.S.D. The alarm required bit is also set and the Alarm level activated.

3.5.5. Store Overflow Interrupt.

The action taken by Executive is almost identical with that taken in 3.5.3. except that the message is 'OUTSIDE STORE LXXX" and the 'STORE OVERFLOW' bit in the P.S.D. is set.

3.5.5. Test Point Interrupt

The level causing the interrupt is inhibited from further action by the manual inhibit bit being set in its P.S.D. and a message is sent to TESTAID, which has the form:-

| | | |
|---|---|---|
| WØ | - | Ø |
| 1 | - | L |
| 2 | - | A |

Where L is the level causing the interrupt

A is the address of the test point instruction.

The rest of the queue item is not used.

### 3.5.6. Executive Call Interrupt

When entered for this reason Executive checks the function, parameter, and data for validity. If valid then action is taken appropriate to the particular call as detailed in section 3.4. If the call is invalid the action taken by Executive depends on whether or not the 'ALL Failures Catastrophic' option is included. In either case the action taken is similar to tha in section 3.5.3. and 3.5.4., the message being "INVALID CALL LXXX' and the level inhibited by the "Invalid Executive Call" bit in the P.S.D.

In the case of both valid and invalid Executive Calls, if the Executive Call Trace option is included and the level concerned has the 'Trace required' bit set in its P.S.D. then a message will be sent to TESTAID and the level inhibited from further action by the manual inhibit bit being set in its P.S.D. The form of the message sent to TESTAID is:-

| | | |
|---|---|---|
| WØ | - | 1 |
| 1 | - | L |
| 2 | - | A |
| 3 | - | NOT USED |
| 4 | - | P |
| 5 | _ | D |

Where L is the level causing the interrupt

A is the address of the Executive Call Instruction

P is the parameter word of the Call

D is the data word of the Call

### 3.5.7. Operator Interrupt

This is caused by operator action only. The interrupt marker inside Executive is incremented, and the 'CALL' level activated.

### 3.5.8. Multi-Level Interrupt Unit Interrupt

The new highest priority level is read from the MIU and if possible it is entered. If it cannot be entered the next highest priority is read until a level is found which can be entered. When an MIU interrupt occurs a check is also performed to ensure that the common interrupt stat in the MIU is set. If this is not the case then the system halts after outputting 'MIU FAILURE' to the teletype. The system can only be re-started by re-loading.

### 3.5.9. System Timer Interrupt

The Timer level for the system is activated, normally level zero. If a non-standard timer is used then this can activate level zero once a second in order to maintain the real-time clock, program timers, device timers, and level reactivation timers on a one second basis.

## 3.6. Exit From Executive

If any level(s) are active and can be entered then Executive exists to the highest priority one, restoring all hardware registers (A,B,P,S, and C) to their values when the level was last active, and also restoring class 2 inihibts to the same state as when the program was last active. This means that in a hardware-levels-only system the lowest level will always be active (since it cannot be reset on the MIU). Consequently this level should de-activate with a WAIT instruction, not an Executive Call. Also, if in a backing-store system (using Space Administr-tion SPADE) it should not have 'Run to completion' set in its P.S.D. as Executive sees the level as being continuously active.

In a system using Software levels all programs must de-activate by means of Executive Calls, as the Software levels administration program always runs on the lowest hardware level and this allows Executive to enter a 'WAIT' state when no level (hardware or software) is active.

# 4.Ø. GENERATION

## 4.1. Options

The following assembly options are available in any combination.

### 4.1.1. Software Levels Option (Option 1).

If more program levels are required than levels on the multi-level interrupt unit (MIU) then this option must be included. The administration of the facility requires the two lowest priority levels of the MIU. There is no difference as far as a program is concerned as to whether it is on an MIU level or a "Software" level except that "Software" levels cannot be directly activated from the hardware, but only via Executive.

### 4.1.2. Segmentation Option (Option 2)

This option has to be included if any program in the system uses overlaying segments, see section 3.2.3.

### 4.1.3. Space Administration (SPADE) Option (Option 3)

This option has to be included if the program overlay facilities are used, see section 3.2.4.

### 4.1.4. System Record Option (Option 4)

This option has to be included if Executive is required to maintain a System Record (see section 3.1.5. and Appendix 5).

The record can be used during testing of multi-level systems to determine the level responsible for a failure.

4.1.5. System Timer Option (Option 5)

This option must be included if any of the following facilities are required:-

Real time clock and calendar

Program timers

Device timers

Timed Program reactivation.

The Time Program must,of course, also be included in the system.

4.1.6. Executive Call Trace Option(Option 6).

This option has to be included if the Executive Call Trace facility in TESTAID is to be implemented. Executive will check whenever a Call is made to see if a Trace is required on the requesting level, and if so a queue item is sent to TESTAID.

4.1.7. Non-Catastrophic Failures option (Option 7)

This option is included to prevent individual program failures halting the machine. If the option is included then the Alarm program will have to be included in order to obtain the appropriate teletype output. If the option is not included then the Alarm program need not be included as Executive outputs the failure message itself.

### 4.1.8. 'Soft' Power Failure Option (Option 8)

If this option is included and a Power failure interrupt occurs then Executive will halt all the IOP's before halting the ALP, as described in section 3.5.1. If the option is not included then the ALP only will be halted, and re-start can only be achieved by re-loading.

### 4.1.9. Non-standard Timer Option (Option 5Ø)

If this option is included then a level specified at assembly or integration time is activated instead of level zero, whenever there is a System Timer Interrupt, as described in Section 3.5.9.

## 4.2. Assembly

The three Executive source tapes must be assembled together, preceeded by an options tape, and followed by a macro call tape of the form:-

```
ELST = P1 = P2 =P3 = P4 = P5 = P6 = P7 = P8 = P9 =
P1Ø = P11 = P12 = P13 = P14 = P15 = P16
ZEND.
```

Where P1 is the total number of hardware levels (i.e. levels on the MIU).

P 2 is the number of additional 'Software' levels required.

P3 is the number of queues required for IOP's and the BIOC (One for each level of the IOP used plus one for each device on the BIOC).

P4 is the number of items allowed in the queue area.

P5 is the priority level assigned to the SPADE program.

P6 is the priority level assigned to the CALL program.

P7 is the priority assigned to the ALARM program.

P8 is the priority level assigned to TESTAID 2.

P9 is the priority level assigned to the TIME program.

P1Ø is the number of times Executive goes through its delay loop between stopping and crash-stopping the IOP's on power failure. The time through the loop is approximately (5S + 4) μs where S is the store cycle time in μs. The minimum value is 1, as zero indicates a count of 65536.

P11 is the number of items allowed in the System Record buffer.

P12 is the start address of the Program Overlay Area.

P13 is the number of blocks in the Program Overlay Area.

P14 is the size (in words) of the Program Overlay Area blocks.

P15 is the number of entries required in the Backing Store Directory.

P16 is the total store size in words.

## 4.3. Assembly Warnings

A list of all the possible assembly warnings is given below, together with a brief description of their cause.

HARDWARE LEVELS EXCEED LIMIT

This warning is output if the number of hardware levels specified exceeds 64, this being the maximum size of the MIU. A default value of 16 is taken.

NUMBER OF HARDWARE LEVELS INVALID

This warning is output if the specifed number of hardware levels is not a multiple of sixteen, or is less than 16, as an MIU must be present for Executive to run, and MIU's are only available in increments of sixteen levels. A default value of 16 is taken.

SOFTWARE LEVELS EXCEED LIMIT

This warning is output if an attempt is made to exceed the permitted maximum number of levels in a system, namely 256. A default value of zero is taken.

I.O.P. LEVELS EXCEED LIMIT

This warning is output if an attempt is made to exceed the total number of queues allowed in a system, namely 1Ø24. A default value of 16 is taken.

ADVISE LARGER QUEUE AREA

The size of the queue area will depend on system considerations, but this warning will always be output if the number of items in the queue area is less than the total number of priority levels (hardware plus software levels).

NUMBER OF OVERLAY AREA BLOCKS INVALID

This warning is output if more than 256 overlay blocks, or less than one,is specified. A default value of two is taken.

START OF OVERLAY AREA INSIDE EXECUTIVE

This warning is output if the start address of the program overlay area is inside Executive. The area is then taken to follow immediately after Executive.

OVERLAY BLOCK SIZE IS NOT A MULTIPLE OF 16 WORDS

The overlay block size must always be a multiple of 16 words so that backing-store transfers always consist of an integal number of sectors. The block size is taken to be the next permitted (i.e. multiple of 16) number.

OVERLAY AREA TOO LARGE

This warning is output if the product of the block size and number of blocks gives an overlay area that is larger than the area available, i.e. the core area not occupied by Executive. No default value is taken, but store overflow failures will occur if an attempt is made to run programs occupying non-existent areas of store.

NUMBER OF POWER FAILURE DELAY LOOPS ZERO OR NEGATIVE

This warning is output because the count is taken as a 16 bit positive number, and a count of zero means the delay loop is obeyed 65536 times.

```
SPADE   LEVEL        XXX   IS INVALID
CALL    LEVEL        XXX   IS INVALID
ALARM   LEVEL        XXX   IS INVALID
TESTAID 2 LEVEL      XXX   IS INVALID
```

The appropriate message is output when the level specified for a particular program is zero, negative or exceeds the levels available. The message will also be output if an attempt is made to assign the program to one of the two lowest hardware levels when software levels are included in the system. This is because these two levels are reserved for software levels administration.

## 4.4. Integration

Integration is not required for the Executive, although it may be expedient to Integrate the Assembler output with other System Programs.

## 5.Ø. Requirements

### Processor

ALP 1 with MIU.

### Store Size

This will be dependant on the combination of options included and the sizes of the various lists. Full details to be announced later.

## 6.O. References

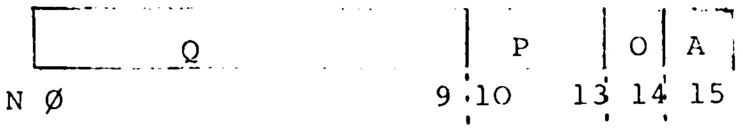
| | |
|---|---|
| Alarm Program | 774Ø2Ø |
| Call Program | 774Ø21 |
| Integrator | 778ØØ2 |
| Space Administration Program | 774Ø3Ø |
| System Record Print Utility | 771Ø4Ø |
| Testaid | 775Ø1Ø |
| Time Program | 774Ø22 |

Appendix 1. Master Segment Structure

Each priority level has associated with it a Master Segment, which is of the form given below.

Word Ø - Storage for register P when program is not running
" 1 - Storage for register S when program is not running
" 2 - Storage for register A when program is not running
" 3 - Storage for register B when program is not running
" 4 - Storage for program conditions register when program is not running
" 5 - Address of Domain zero for the program
" 6 - Number of channels available to the program (C)
" 7 - Number of segments available to the program (S)
" 8 - Value (in seconds) of the program timer.

The following (C) locations specify the channels available to the program. The format of each word is:-



Where Q is the destination queue number
P is the message priority (see section 2.1.9.)
O is the Over-ride Allocation bit which if set prevents the channel allocated bit from being reset.
A is the channel allocated bit which is set by Executive when a channel is allocated, and reset when the channel is de-allocated, provided that the Allocation Over-ride bit is not set.

One such item occurs for every channel available to the program, and the channels are numbered , in the order in which they are defined, from 1 to (C).

The next (2S-3Ø) locations contain the segment definitions in numerical order from 16 to S, each definition consisting of two words as follows:-

| N | Ø | 1 | 2 | 3 | 15 |
|---|---|---|---|---|---|
| Word Ø | X | Y | Z | BSR | |
| Word 1 | SA | | | | |

Where X is set, if the segment is currently on backing-store

Y if set, indicates that a clean copy of the segment is required when it is next requested.

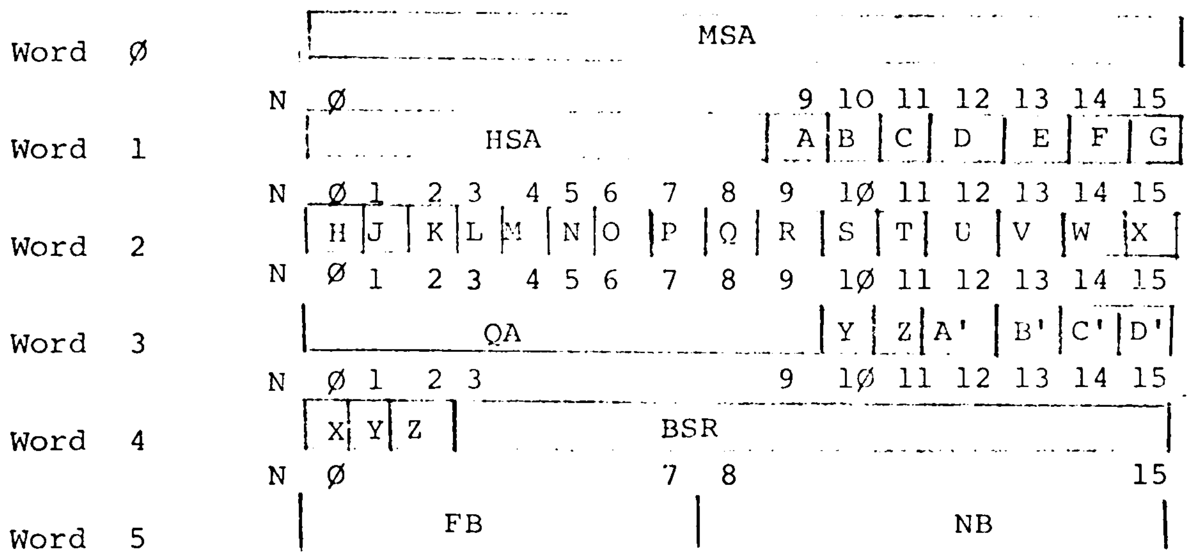
Z if set, indicates that both clean and dump areas exist on backing store for the segment

BSR this is the number of the backing-store directory slot for the dump area of the segment. The clean copy slot if it exists, will always be immediately following.

SA is the first location in core occupied by the segment.

## Appendix 2 Program Status Definition List Structure

Each program in the system has one entry in this list consisting of four words. In systems which have a program overlay area an additional two words per program are required. The entries are in priority level order, and are as follows.


Word Ø
MSA
N Ø 9 10 11 12 13 14 15
Word 1
HSA
A B C D E F G
N Ø 1 2 3 4 5 6 7 8 9 1Ø 11 12 13 14 15
Word 2
H J K L M N O P Q R S T U V W X
N Ø 1 2 3 4 5 6 7 8 9 1Ø 11 12 13 14 15
Word 3
QA
Y Z A' B' C' D'
N Ø 1 2 3 9 1Ø 11 12 13 14 15
Word 4
X Y Z
BSR
N Ø 7 8 15
Word 5
FB
NB

Only present Program Overlay Used.

Where MSA is the start address of the program's Master Segment

HSA is a count of the number of data transfers a program in the overlay area has currently in progress.

A if set, indicates that the program does not use the program overlay area.

B is set whenever a program in the program overlay area has any data transfers in progress.

C if set, prevents a program in the overlay area being dumped before it has exited : either normally or because it is waiting for a queue item, allocation of a channel, or queue space, or has attempted to perform too many transfers.

D is reserved for future expansion.

E is set when the program is running, waiting to run, or waiting for a segment.

F is set if an Executive Call Trace is currently required for the program.

G is set when Space Administration is performing a transfer for the program.

H is reserved for use by Testaid.

J is set if a level has been activated but the program inhibit bit (Q) was set. When entry is allowed to the level the bit is reset and the level activated.

K is set if the program is waiting for Space Administration to bring another program into core from backing-store, when a message with data associated is being sent.

L is set when the queue associated with the program is allocated to a program.

M is set when another program requires allocation of this program's queue.

N is set if the program's job segment is currently occupied.

O is reserved for future expansion
[illegible]

P is set when the program has failed(because of an invalid function,etc) and indicates to the Alarm program that a message needs to be output on the teletype for this program.

Q is set when a program is inhibited from action due to an Inhibit Executive Call.

R is set when a program is inhibited from action due to Testaid action.

S is reserved for future expansion.

T is set if there has been a failure of the backing-store area used to store any of the program.

U is set if the program has caused a store overflow violation (i.e. tried to access outside the available store).

V is set if program has 'timed out' (i.e. taken over the prescribed time limit for dealing with a job).

W is set if the program has attempted to obey an invalid instruction.

X is set if the program has attempted to make an Executive Call which is invalid.

QA is only valid if either Z or B' is set and is the queue or segment which the program is waiting for.

Y is set if the program attempted to send more than its permitted number of simultaneous transfers.

Z is set if the program has performed a Hold for Queue Item Executive Call. The Queue (not the channel) for which the program is waiting is in QA.

A' is set if the program is waiting for allocation of a queue

B' is set if the program is waiting for Space Administration to bring a segment into core. The segment number required is in QA.

C' is set if the program is waiting for space to become available in the queue area in order to send a message.

D' is set if the program is not complete in core, and is waiting to be brought into core by Space Administration.

The remainder of the Program Status Definition is only required if a Program Overlay Scheme is included in the system.

X' is set when the program is on backing store.

Y' is set if a clean copy of the program is required the next time it is brought into core.

Z' is set if both clean and dump areas exist for the program on backing store.

BSR is the number of the Backing store directory slot for the program dump area. The clean copy slot, if it exists is always the one immediately following.

FB is the number of the first block in the program overlay area which the program occupies (numbered from zero).

NB is the number of blocks in the program overlay area which the program occupies.

The Program Status Definition List commences at label ZSPD.

## Appendix 3 Queue Definition List Structure.

This is a three word per item list, consisting of two types of item. The first part of the list has one entry per program level, with the following format.

Word Ø - The address of the first item in the program's queue.

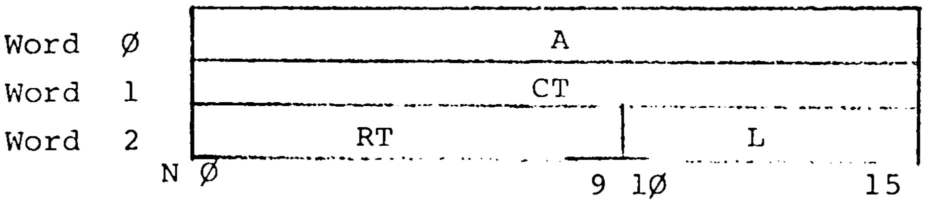
Word 1 - The current value of the program's timer ( in seconds).

Word 2 - The number of simultaneous transfers still available to the program.

The queue definition for queue zero does not refer to program level zero's queue, but to the queue of unused items in the queue area.

If the address of the queue item (word zero) is zero, this implies that the queue is empty.

The second part of the list has one entry per IOP level, and the format is:-



Where A is the address of the first item in the queue (zero if empty).

CT is the current value of the transfer·timer.

RT is the reset value of the transfer tiner.

L is the program level occupied by the IOP control program.

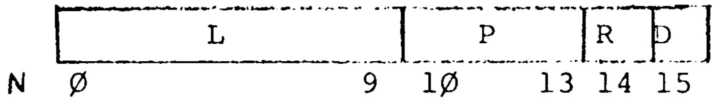
The Queue Definition List commences at label ZQDEF.

## Appendix 4 Queue Area Structure

This is an area of core in which all the queues for program levels and IOP levels are kept. Each queue is in the form of chained items, allowing greater flexibility in the size of individual queues than a fixed size for each queue. Queue items have the following format:-

Word Ø - Address of next item in queue (zero if no more items)

Word 1



Where L is the sending program level

P is the message priority

R is set if the item is to be returned

D is set if the queue is de-allocated.

Words 2-7 are a copy of the sending program's message, with word 5 (the status word) set to zero. If the user made a de-allocate call then words 3 to 7 are not used and word 2 contains "all ones".

## Appendix 5 System Record Area Structure

This consists of an area of core used as a cyclic buffer, each item consisting of two words.

If bit zero (NØ) of word zero is zero then the item is a record of an Executive Call and has the following format:-

| N | Ø | 1 6 | 7 | 8 15 |
|---|---|---|---|---|
| Word Ø | Ø | F | I | L |
| Word 1 | A | | | |

Where F is the least significant six bits of the function field of the Executive Call, (the most significant bit is always zero).

I is the indirect marker for the parameter of the Executiv Call.

L is the program level which made the Call.

A is the address of the Executive Call instruction.

If bit zero of word zero is set and bit one is zero then the item i the record of a program level being entered, and has the following format:-

| N | Ø | 1 | 2 7 | 8 15 |
|---|---|---|---|---|
| Word Ø | 1 | Ø | N/U | L |
| Word 1 | A | | | |

Where L is the program level entered

A is the address at which the program was entered.

If bits zero and one of word zero are both set then the item is the record of an attempt to enter a level which was inhibited from actior The format is :-

| N | Ø | 1 | 2 7 | 8 15 |
|---|---|---|---|---|
| Word Ø | 1 | 1 | PSD (Y-D') | L |
| Word 1 | X' | | PSD (A-G) | PSD (Q-X) |

Where L is the program level which could not be entered

X' is set if the program was on backing-store (as bit X' of the Program Status Definition).

PSD (Y-D'), PSD (A-G) and PSD (Q-X) refer to bits (Y-D') etc. of the Program Status Definition (see Appendix 2).

The System Record Parameters are stored in three core locations starting at label ZDIAG. The contents of the locations are:-

| | | |
|---|---|---|
| Word Ø | - | First word of oldest item in the Record. |
| Word 1 | - | Start address of the Record Area. |
| Word 2 | - | Address of the first location beyond the Record area. |

## Appendix 6 Executive Calls - Macro Form

The formats for an Executive Call in the macro form are:-

1) MNEM = P=D
2) MNEM = P=I=D
3) MNEM = P=D=I
4) MNEM = P=I=D=I

where MNEM is a mnemonic for the particular function required.

P is the parameter displacement.

D is the data displacement.

(1) is used where both parameter and data are accessed directly.

(2) is used where the parameter is accessed indirectly and the data directly.

(3) is used where the parameter is accessed directly and the data indirectly.

(4) is used where both the parameter and the data are accessed indirectly.

The mnemonics corresponding to the Executive Calls given in section 3.4. are given below.

EXIT - Exit from Program (section 3.4.1.)
ABND - Abandon Program (section 3.4.2).
DATE - Request Date (section 3.4.3.)
TIME - Request Time (section 3.4.4.)
ACTV - Activate Another Program (section 3.4.5.)
INHE - Inhibit Entry to a Program (section 3.4.6.)
ALLE - Allow Entry to a Program (section 3.4.7.)
GOXX - Send Message (section 3.4.8.)
GOXA - Send Message and Allocate (section 3.4.9.)
GORX - Send Message and Return (section 3.4.1Ø.)
GORA - Send Message, Return and Allocate (section 3.4.11.)
HOLD - Hold for Queue Item (section 3.4.12.)
QUEA - Test for Queue Available (section 3.4.13.)
DEAL - De-Allocate Channel (section 3.4.14.)

CAAE - Call Segment, Abandoning Current Segments and Enter
(section 3.4.15.)

CADE - Call Segment, Dumping Current Segments and Enter
(Section 3.4.16).

CAAR - Call Segment, Abandoning Current Segments and Return
(section 3.4.17.)

CADR - Call Segment, Dumping Current Segments and Return
(section 3.4.18.)

SEGA - Test for Segment Available (section 3.4.19.)

## Appendix 7. Invalid Executive Calls

The list below gives reasons for a level being terminated due to an invalid Executive Call.

1) An unused Executive Call function was specified.

2) The Call refers to a non-existent channel.

3) The Call refers to a non-existent segment.

4) The Call refers to a segment whose backing-store directory entries have not been set up.

5) A Call which attempts to abandon anything other than a segment currently in core.

6) A Call which attempts to send a message with priority fifteen when the sending program has nothing in its job segment.

7) Any Call which refers to a channel with an invalid queue number specified in it.

8) Any Call to activate a program, inhibit entry or allow entry to a program which specifies channel zero.